

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-195301

(43)Date of publication of application : 19.07.2001

(51)Int.Cl.

G06F 12/08

G06F 9/46

(21)Application number : 2000-001642

(71)Applicant : JAPAN ADVANCED INST OF
SCIENCE & TECHNOLOGY
HOKURIKU

(22)Date of filing : 07.01.2000

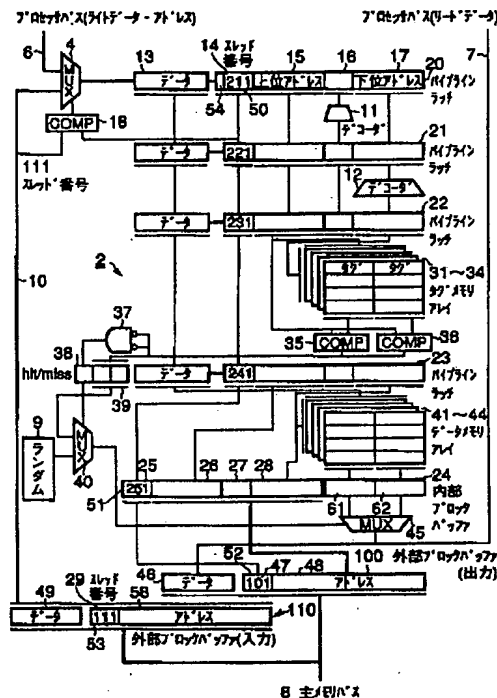
(72)Inventor : HIBINO YASUSHI
UKAI KAZUTOSHI

(54) UPDATING CIRCUIT FOR PIPELINE CACHE MEMORY

(57)Abstract:

PROBLEM TO BE SOLVED: To evade the contention of the updating processing of data and an access processing from a succeeding thread at the time of a cache mishit and to prevent the disturbance of a pipeline operation.

SOLUTION: This circuit is provided with a first pipeline latch (20) to be the interface of a processor (1) and a cache memory (2), a second pipeline latch (21) arranged in an intermediate stage for constituting a pipeline and an external block buffer (110) to be the interface of a main memory and the cache memory. The first pipeline latch is provided with fields (50) and (54) for displaying a thread identifier and the field for displaying a thread state and the external block buffer is provided with the field for displaying the thread identifier, a block address and an intra-set block number. Then, a comparator circuit (18) for judging by comparing the thread identifier displayed by the first pipeline latch with the thread identifier displayed by the external block buffer is provided.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号 ✓
特開2001-195301
(P2001-195301A)

(43) 公開日 平成13年 7月19日 (2001.7.19)

(51) Int.Cl.⁷

G 0 6 F 12/08

識別記号

9/46

3 4 0

F I

G 0 6 F 12/08

9/46

テーマコード(参考)

Q 5 B 0 0 5

B 5 B 0 9 8

3 4 0 B

審査請求 有 請求項の数 6 O L (全 7 頁)

(21) 出願番号 特願2000-1642(P2000-1642)

(22) 出願日 平成12年 1月 7日 (2000.1.7)

(71) 出願人 394023953

北陸先端科学技術大学院大学長

石川県能美郡辰口町旭台 1丁目 1番地

(72) 発明者 日比野 靖

石川県能美郡辰口町旭台 1-50 A44

(72) 発明者 齋藤 和哉

兵庫県加古川市加古川町北在家720-10

(74) 代理人 100058479

弁理士 鈴江 武彦 (外 5 名)

Fターム(参考) 5B005 JJ12 KK12 KK23 MM01 NN43

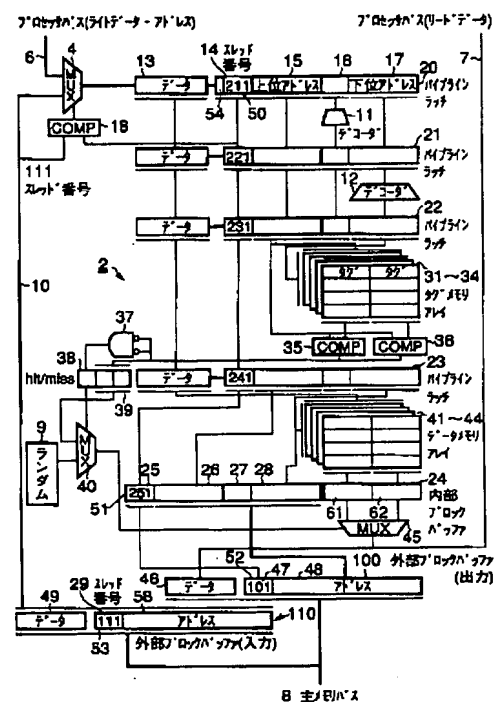
5B098 AA02 AA03 GA05

(54) 【発明の名称】 パイプラインキャッシュメモリの更新回路

(57) 【要約】

【課題】 キャッシュミスヒット時に、データの更新処理と、後続のスレッドからのアクセス処理との競合を回避し、パイプライン動作の乱れを防止する。

【解決手段】 プロセッサ(1)とキャッシュメモリ(2)とのインタフェースとなる第1のパイプラインラッチ(20)と、パイプラインを構成する中間ステージに配置された第2のパイプラインラッチ(21)と、主メモリとキャッシュメモリとのインタフェースとなる外部ブロックバッファ(110)とを具備し、第1のパイプラインラッチにはスレッド識別子とスレッド状態を表示するフィールドとを表示するフィールド(50)、(54)が設けられ、外部ブロックバッファにはスレッド識別子、ブロックアドレス、およびセット内ブロック番号を表示するフィールドが設けられ、そして、第1のパイプラインラッチの表示するスレッド識別子と、外部ブロックバッファの表示するスレッド識別子とを比較判定する比較回路(18)を具備する。



【特許請求の範囲】

【請求項 1】 プロセッサと主メモリと間に配置され、プロセッサから要求される複数のスレッドに関して複数のステージでパイプライン処理を実行するようにパイプライン構成されたキャッシュメモリにおいて、前記プロセッサと前記キャッシュメモリとのインタフェースとなる第 1 のパイプラインラッチと、パイプラインを構成する中間ステージに配置された第 2 のパイプラインラッチと、前記主メモリと前記キャッシュメモリとのインタフェースとなる外部ブロックバッファとを具備し、

前記第 1 のパイプラインラッチは処理すべきスレッドを識別するスレッド識別子とこのスレッドの状態を表示するフィールドとを含み、前記第 2 のパイプラインラッチはこのステージで処理されるスレッド識別子を表示するフィールドを含み、前記外部ブロックバッファは該ブロックに対応するスレッド識別子、ブロックアドレス、およびセット内ブロック番号を表示するフィールドを含み、そして、前記第 1 のパイプラインラッチの表示するスレッド識別子と、前記外部ブロックバッファの表示するスレッド識別子との一致、不一致を比較判定する比較回路とを具備することを特徴とするキャッシュメモリ更新回路。

【請求項 2】 前記キャッシュメモリはセットアソシアティブキャッシュメモリであることを特徴とする請求項 1 に記載のキャッシュメモリ更新回路。

【請求項 3】 前記スレッドの状態を表示するフィールドには、前記キャッシュメモリが該当するスレッドに関しミスヒットの場合に休止中である旨表示される請求項 1 に記載のキャッシュメモリ更新回路。

【請求項 4】 前記比較回路が一致と判定した場合に、前記プロセッサの制御により、該当するスレッドに係るデータ更新処理が実行される請求項 1 に記載のキャッシュメモリ更新回路。

【請求項 5】 データ更新時におけるセット内ブロック番号の選択のためのランダム発生器を具備することを特徴とする請求項 2 に記載のキャッシュメモリ更新回路。

【請求項 6】 前記第 1 のパイプラインラッチおよび第 2 のパイプラインラッチは、読み出し要求のあったデータのアドレスをデコードする各デコーダを介して次のステージに接続されていることを特徴とする請求項 1 または請求項 2 に記載のキャッシュメモリ更新回路。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明はキャッシュメモリがミスヒットした際のデータブロックの更新回路に関するもので、特にパイプライン構成のキャッシュにおける更新回路に関するものである。

【0002】

【従来の技術】 コンピュータのメモリシステムは、大容

量、高速および低価格の 3 つの要求を満たす必要があるため、総てを 1 種類のメモリで実現することは不可能である。このため、容量と速度の異なる複数のメモリを階層的に組合わせたメモリシステムが採用されている。

【0003】 キャッシュメモリは、プロセッサと主メモリの速度差を埋めるためにプロセッサと主メモリの中間に配置された高速、小容量のメモリである。プログラムの局所性を利用して、最近利用された命令またはデータを格納しておき、メモリのアクセス性能を向上させる。

【0004】 このため、通常のキャッシュメモリはアクセス時間が短いことが第一であり、パイプライン構造の採用は積極的には行われていない。せいぜい、アドレスコードと、メモリアレイ本体との 2 段のパイプライン構成が採用されている程度である。

【0005】 しかし、マルチスレッドプロセッサなど特殊なアーキテクチャを対象にした場合のキャッシュメモリは、アクセス時間ではなくスループットが問題となる場合がある。つまり、レイテンシ(時間遅れ)は大きくてもよいが、出来るだけ短いサイクル時間にデータを読み上げることが重要となり、かかるプロセッサにおいては本質的にパイプライン構造が適している。

【0006】 例えば、マルチスレッド型プロセッサにおいては、キャッシュメモリへのアクセス要求は、毎サイクル、異なるスレッドからの要求であり、その要求のアクセス時間には余裕があるので、キャッシュメモリをパイプラインで構成することができる。

【0007】 しかしながら、パイプライン構成のキャッシュでは、キャッシュアクセスがヒットのときは問題ないが、ミスヒット時に、キャッシュエントリの更新(アップデート)処理と、後続のスレッドからのアクセス処理とを同時並行に行なわなければならない、主メモリアレイへのアクセスが競合してパイプライン動作が乱れ、スループットが低下してしまうという問題点があった。

【0008】

【発明が解決しようとする課題】 したがって、本発明はキャッシュメモリにパイプライン構造を適用した場合に生ずる上記問題点に鑑みてなされたものであり、キャッシュミスヒット時のキャッシュエントリの更新処理と、後続のスレッドからのアクセス処理との競合を回避し、パイプライン動作の乱れを防止し、スループットが低下しないようにすることをその目的とする。

【0009】

【課題を解決するための手段】 本発明は、かかる目的を達成するよう、プロセッサと主メモリと間に配置されプロセッサから要求される複数のスレッドに関して複数のステージでパイプライン処理を実行するようにパイプライン構成されたキャッシュメモリにおいて、前記プロセッサと前記キャッシュメモリとのインタフェースとなる第 1 のパイプラインラッチと、パイプラインを構成する中間ステージに配置された第 2 のパイプラインラッチ

と、前記主メモリと前記キャッシュメモリとのインタフェースとなる外部ブロックバッファとを具備し、前記第 1 のパイプラインラッチは処理すべきスレッドを識別するスレッド識別子とこのスレッドの状態を表示するフィールドとを含み、前記第 2 のパイプラインラッチはこのステージで処理されるスレッド識別子を表示するフィールドを含み、前記外部ブロックバッファは該ブロックに対応するスレッド識別子、ブロックアドレス、およびセット内ブロック番号を表示するフィールドを含み、そして、前記第 1 のパイプラインラッチの表示するスレッド識別子と、前記外部ブロックバッファの表示するスレッド識別子との一致、不一致を比較判定する比較回路とを具備するキャッシュメモリ更新回路である。

【0010】そして、本発明においては、前記キャッシュメモリはセットアソシアティブキャッシュメモリであることを可能とし、前記スレッドの状態を表示するフィールドには前記キャッシュメモリが該当するスレッドに関しミスヒットの場合に休止中である旨表示されることを可能とし、前記比較回路が一致と判定した場合に前記プロセッサの制御により該当するスレッドに係るデータ更新処理が実行されることを可能とする。

【0011】また、本発明においては、データ更新時におけるセット内ブロック番号の選択のためのランダム発生器を具備することを可能とし、前記第 1 のパイプラインラッチおよび第 2 のパイプラインラッチは読み出し要求のあったデータのアドレスをデコードする各デコーダを介して次のステージに接続されていることを可能とする。

【0012】

【発明の実施の形態】本発明の実施の形態を以下に図面を参照して説明する。本発明においては、キャッシュメモリ 2 のパイプラインラッチ（パイプラインステージの情報を保持するレジスタ）20、21、22、23に、アドレスフィールド 15～17 およびデータフィールド 13 と共に、スレッド識別子を格納するフィールド 50 とスレッドの状態（活動中、休止中）を表示するフィールド 54 を設ける。

【0013】かかるスレッドの状態を表示するフィールド 54 を設けるのにより、ヒットしたスレッドかミスヒットのスレッドかが表示され、そのスレッドがミスヒットであることがわかる。そして、例えばミスヒットであった場合でも、ミスヒットのあったスレッドの各処理はパイプラインの各ステージにおいてそれぞれ中止されるが、パイプライン内においてはミスヒットのあったスレッドを含めて連続したパイプライン処理がそのまま続行される。

【0014】一方、ミスヒットのあったスレッドについては、主メモリ 3 から更新のためのデータを読み出してそのスレッドの識別子と共に外部ブロックバッファ 110 に格納する。ミスヒットの更新処理は、まず、キャッ

シュ部入口に配置された比較器すなわち比較回路 18 による比較により、データ更新のために主メモリ 3 から読み出されそして外部ブロックバッファ 110 に格納されたデータに係るスレッドの識別子 29 と、パイプラインで連続処理されているスレッド識別子 14 とが比較される。そして、この比較の結果、双方のスレッド識別子一致した場合のみに、処理が中止されたそのスレッド用のパイプラインサイクル（空きサイクルという）を使用して他のスレッドの処理を乱すことなく連続的に更新処理される。

【0015】すなわち、上記外部ブロックバッファ 110 には該当するアドレスが格納されているのでこのアドレスによりアクセスし、データフィールド 49 に格納されたデータによって先にミスヒットしたタグメモリアレイ 31～34 の内容および対応するデータメモリアレイ 41～44 のデータブロック内容のデータ更新が行なわれる。

【0016】図 1 に示すパイプラインキャッシュ 2 は、多段に構成されたアドレスデコーダ 11、12 とパイプラインの各段毎に挿入されたパイプラインラッチ 20、21、22、23 を有する。パイプラインラッチ 20 はプロセッサ 1 とキャッシュメモリ 2 とのインタフェースとなり、パイプラインラッチ 21、22、23 はパイプラインを構成する各中間ステージに配置される。

【0017】このパイプラインキャッシュはさらに、複数のタグメモリアレイ 31、32、33、34 と、同じく複数のデータメモリアレイ 41、42、43、44 と、タグ比較を行なう複数の比較器 35、36 と、複数の内部読みだしブロックバッファデータフィールド 61、62 とを有し、さらに、1 語幅のプロセッサバス 7 と、ブロックサイズ幅の主メモリバス 8 と、ブロック選択のためのランダム発生器 9 と、外部読みだしブロックバッファ 100、110 とスレッド識別子の一致判定回路である比較器 18 を含む。

【0018】また、パイプラインラッチ 20、21、22、23、内部ブロックバッファ 24 および外部ブロックバッファ 100、110 には、データフィールド 13、46、49 およびアドレスフィールド（15、16、17）、（26、27、28）、48、51 を設ける。それと共に、各スレッドを識別するスレッド番号すなわちスレッド識別子（図 1 において例えば 211、221、231、241、251、101、111 のスレッド番号により示される。）を格納するフィールド 50、51、52、53 を設け、さらに、ミスヒットの場合に空きサイクルであることを示すフィールド 54 を設ける。

【0019】データメモリアレイ 41～44 の容量が増えた場合には、デコードの階層を増やし、パイプラインの段数を増やすことが当然考えられる。図 1 のスレッド番号 211、221、231、241、251 は、ある

10

20

30

40

50

時点においてパイプラインの格ステージでこれら表示されたスレッドに係る処理が平行して行われていることを示す。なお、図1はセットアソシアティブキャッシュメモリの例を示すが、本発明はセットアソシアティブキャッシュメモリに限定されるものではない。

【0020】図1の実施の形態においては、外部ブロックバッファ110にもブロックアドレスを格納するアドレスフィールドを設けている。このため、通常のキャッシュメモリのように入力（リード）要求したブロックのアドレスを記憶しておく場所をキャッシュメモリ2内に設ける必要がない。即ち、リード要求と共に主メモリ3にアドレスを送り出し、該アドレスが更新データと共にキャッシュメモリ2に戻ってくる構成を採用しているので、主メモリに対する応答が戻ってくるまでの間キャッシュメモリをウェイト状態にしてアドレスを保持しておく必要がない。このため各スレッドに対するパイプライン処理が継続して実行できる。さらにアドレスフィールドにはセットアソシアティブ方式の採用に対応し、セット内ブロック番号を格納するフィールドが設けられている。

【0021】hit/misの判定回路であるNAND回路37を有し、この回路37はタグメモリアレイ31～34の出力側にある2つの比較器35、36の出力の否定のANDをとる。したがって、両方の比較器が不一致を出力した場合にエントリが存在しない（ミス）ということになる。NAND回路37に接続されたラッチのフィールド38はこのhit/misの判定結果を記憶し、フィールド39は両方の比較器35、36の比較結果を記憶する。

【0022】hit/misの判定結果はプロセッサ1に接続された経路を介してプロセッサに戻される（図示せず）。プロセッサ1は自分のメモリ（図示せず）に各スレッド識別子に対応させてこの判定結果を例えばテーブル形式で記憶し、ミスの判定のあったスレッドに関しては、空きサイクルである旨の表示と共にスレッド識別子をキャッシュメモリ2に送り、これらはパイプラインラッチ20に格納される。

【0023】プロセッサ1とキャッシュメモリ2との間は、図2（a）に示すように、1語幅のプロセッサバスにより接続される。キャッシュメモリ2と主メモリ3との間は1ブロック幅の主メモリバスにより接続される。図2（b）にプロセッサバスを7通るデータの構成を、図2（c）に主メモリバス8を通るデータの構成を示す。

【0024】図1の実施の形態においては、全体のパイプライン段数は5段とし、アドレスデコードのパイプライン段数は2段、セットアソシアティブの連想度は2、タグの比較器35、36の数は2としているが、一般にはさらに多くのパイプライン段数とすること、あるいは連想度を増やすことが可能である。上記セットなる用語は、タグとデータブロックの対を要素として、連想度の

数だけ集まったものを意味する。

【0025】以下（a）ヒット時のリードオペレーション、（b）ミスヒット時のリードオペレーション、（c）ライトオペレーション（ヒット時）および（d）ライトオペレーション（ミスヒット時）に分けてこの実施の態様の動作を説明する。

【0026】（a）ヒット時のリードオペレーション
まず通常のリードオペレーションを説明する。プロセッサ1から読み出し要求を受け、パイプラインラッチ20に格納されたアドレスは、スレッド識別子211と共にアドレスデコーダ11、パイプラインラッチ21、アドレスデコーダ12、パイプラインラッチ22を通り、タグメモリアレイ31～34に到達する。

【0027】上記アドレスのうち下位アドレス（16、17に格納）はアドレスデコーダ11および12においてそれぞれデコードされる。すなわち、本実施の形態においては、図1に示すように、階層的デコードを行なうためおよび以下に述べる理由により、デコーダを直列的に接続されるデコーダ11とデコーダ12とに分けている。各デコーダの論理は通常のデコーダの論理と同じであり、 n 本の信号かつ、 2^n （ 2 の n 乗）の信号に分解するものである。すなわち、下位アドレスをデコードするデコーダを複数のデコーダ11、12により形成し、一方下位フィールドを複数のメモリ小アレイ16、17に分配し、そこでそれぞれ用のデコーダによってデコードするようにしたものである。

【0028】このように下位アドレスのフィールドを分割してこれらを階層的にデコードすることにより、1)フィールド長が短くなるためデコード時間が短縮され、さらに2)デコーダを小アレイに分散配置することにより、配線遅延を短縮できる。

【0029】デコーダ11および12によりデコードされた下位アドレスにより、タグメモリアレイ31～34から下位アドレスに該当するタグが読み出される。読み出されたタグは、上位アドレスのビットと比較器35、36で比較される。これら比較器のうちいずれかが「等しい」と判定すればすなわちヒットであるので、パイプラインラッチ23にラッチされているアドレスにより、データメモリアレイ41～44をアクセスする。データメモリアレイ41～44はアクセスが完了すると読み出されたデータを内部ブロックバッファ24のデータフィールド61、62にラッチし、プロセッサバス7を通じてプロセッサ1に送信する。

【0030】以上の動作は、図3（a）に示すように、（1）デコーダ11によるアドレスデコード1、（2）デコーダ12によるアドレスデコード2、（3）タグ読み出し、およびタグと上位アドレスとの比較、（4）データメモリアレイ41～44にアクセスしデータ読み出し、（5）プロセッサバス7を介してのプロセッサへのデータ送り出しの合計5段のパイプライン処理で行なわ

れる。このよう構成し動作させることにより、クロックサイクル毎に新たなアドレスを受け付け、各5クロック後に連続してキャッシュデータを得ることができる。

【0031】(b) ミスヒット時のリードオペレーション

ミスヒット時のリードオペレーションを、ミスヒットの検出、更新ブロックの選択、ダーティブロックのライトバック、更新ブロックの読み出し、およびタグとブロックの更新に分けて以下説明する。

【0032】b-1 ミスヒットの検出

まず、リードオペレーションのときにミスヒットが生じた場合を説明する。この場合、パイプライン処理における上記(3)のタグアクセスおよびタグと上位アドレスとの比較のステージまでは同様である。

【0033】比較器35、36の出力が共に等しくないとされた場合、すなわちミスヒットの場合であるが、この場合もヒット時と同様にメモリアccessを行ない、読み出したデータをタグと共に内部ブロックバッファ24のデータフィールド61〜62にラッチする。この場合のデータは、キャッシュアドレス(下位のアドレス)によって読み出されたタグとデータの対である。

【0034】タグが一致しないのでミスと判定されるだけで、データメモリアレイ41〜44の読み出しはミス、またはヒットにかかわらずまったく同じように行われる。読み出されたデータはそのままプロセッサ1にも送られるが、プロセッサ1はスレッド識別子と共にミスヒットである判定結果を受信しているので送られたデータを破棄する。ヒット時と異なる点は、プロセッサ1にミスヒットであることをスレッド識別子によって伝えることである。

【0035】プロセッサ1は、ミスヒットとなったアクセス要求を出したスレッドをスレッド識別子によって判定して、該当するスレッドを空きサイクルにする。すなわち、空きサイクルとなったスレッドはキャッシュメモリ2に対し空きサイクルであるという指示と共にスレッド識別子を送る。この指示を受け、該当するスレッドについては、パイプラインラッチ20のスレッドの状態(活動中、休止中)を表示するフィールド54は休止中であることを表示する。

【0036】フィールド54に空きサイクルを示す休止中の表示がある場合、キャッシュメモリ2はこのスレッドに関しては該当する各ステージにおいて実際のアクセス動作(デコーダによるアドレスデコード、タグアクセスおよびタグと上位アドレスとの比較、メモリアレイアクセス、データ送り出し)を行わない。プロセッサ1は、空きサイクルのスレッド以外のスレッドからは(特に命令キャッシュの場合)キャッシュメモリ2に毎サイクルアクセス要求が出される。

【0037】b-2 更新ブロックの選択

ミスヒットとなったエントリは、更新を行なう必要があ

る。更新は、休止中のスレッドによる空きサイクルを使って行なう。

【0038】まず、更新するブロックのデータを主メモリ3に追い出す。この場合、更新するブロックを選択(セット内ブロック番号)する必要がある。この選択においては通常のキャッシュメモリにおける選択のように、使用されずにいた時間が最も長いブロックを置き換えるいわゆるLRU(least recently used)法を使うことは困難である。

10 【0039】なぜなら、パイプラインキャッシュメモリでは、毎サイクルすべてのステージでの動作がある。また、毎サイクル、タグメモリアレイ31から34およびデータメモリアレイ41から44にアクセスが行なわれる。従って、LRUスタックを更新するための時間をとることが困難であり、LRU法によっては高速なサイクル時間を実現することができないからである。このため、LRU法に代わる方法を採用する必要がある。

20 【0040】本実施の形態では、ランダムアルゴリズムを採用している。ランダムアルゴリズムは比較的よい性能を示す。すなわちブロックの選択(セット内ブロック番号)はランダム発生器9の出力により決定する。セット内ブロック番号は、キャッシュミスのあった場合に追い出しの選択されたブロックについて、セットを構成するタグとデータブロックの対に番号をつけたものである。例えば、n個のセットからなる連想度2のキャッシュメモリにおいては、セット内ブロック番号は0または1で規定される。

30 【0041】一般に、選択されたブロック(セット内ブロック番号)をAとすると、このAの決定は事前に行なっておけばよい。すなわち、事前にランダム発生器9により生成しておいた値を得ればよい。ランダム発生器9は次の要求に備えて次のランダム数を発生するよう動作する。

【0042】なお、このブロック番号を表示するブロックアドレスは、通常下位アドレスの上位ビットで指定され、下位アドレスの下位のビットはブロック内の語を選択する。

【0043】b-3 ダーティ(dirty)ブロックのライトバック

40 追い出しブロックを決定した後、選択された追い出しブロック内のデータを実際に主メモリにライトバックする必要がある。すなわち、ブロックがダーティである場合(ダーティ; 書き込みが行われたキャッシュの内容と主メモリの内容が一致しない)、追い出しブロックのデータを記憶している内部ブロックバッファ24の内容を主メモリバス8を通して、1クロックサイクルで主メモリ3へと送り出す。

50 【0044】主メモリ3への書き込みアドレスは、内部ブロックバッファ24のアドレス記憶フィールド26、27、28に記憶されたタグおよび下位アドレスより生

成できる。数サイクルの書き込み時間の後に、主メモリ 3 への書き込み動作は完了する。

【0045】b-4 更新ブロックの読み出し

キャッシュメモリのデータ更新のため、主メモリへの上記書き込み動作の完了をまたずに、要求アドレス、スレッド識別子、セット内ブロック番号の 3 つ組により、主メモリに対しリードアクセス要求を行なう。要求後数サイクルの後に主メモリ 3 に対するアクセスが完了する。

【0046】主メモリから読み出されたブロックは、リードアクセス要求時のアドレス、スレッド識別子、セット内ブロック番号の 3 つ組に、データブロックを付け加えた 4 つ組として、主メモリバス 8 を通じて送られ、外部ブロックバッファ 110 に蓄えられる。その後、図 1 の左側のライン 10 (太線にて表示) を通じてキャッシュ部入口のマルチプレクサ 4 まで送られ、図 3 (c) に示すパイプライン処理工程を経て、最終的にデータメモリアレイ 41~44 に書き込まれる。

【0047】b-5 タグとブロックの更新

該当するスレッド識別子の移動に対応して、図 3 (c) に示すようにデコーダ 11 およびデコーダ 12 による各処理工程 (デコード 1、デコード 2) が行われる。その後、まずタグメモリアレイ 31~34 にアクセスし、タグをアドレスとセット内ブロック番号により更新する。つぎにデータブロック内容をアドレスとセット内ブロック番号によりデータメモリアレイ 41~44 の該当ブロックに書き込み、エントリの更新を行なう。

【0048】この更新のための書き込みタイミングは、同様に外部ブロックバッファ 110 に記憶されているスレッド番号 (111) と、パイプラインのメモリアレイステージを流れているスレッド識別子のスレッド番号が一致するときである。このとき同時に該当ブロック内のダーティビットをクリアする。更新の終了したこの時点で、プロセッサ 1 は、ミスしたスレッドの休止状態を活動状態へ戻す。

【0049】なお、更新の場合には 5 段目のステージの動作 (図示せず) は通常は「空」となりクロックのみを進める。もし何らかのデータがプロセッサ 1 に送られた場合には、プロセッサ 1 はそのデータは受け取らずに捨てる。このように、データメモリアレイ 41~44 の該当ブロックへの書き込みは、休止中のスレッドの空きサイクルに行なうので、他の活動中スレッドのメモリアレイアクセスと競合することはない。

【0050】主メモリ 3 に対する書き込み要求にスレッド識別子を付与することにより、キャッシュメモリ 2 から主メモリ 3 への書き込み要求が非同期的に行なわれ、大きなレイテンシの後にアクセスが完了するような場合においても、パイプライン処理との同期を正確にとることができる。

【0051】また、セット内ブロック番号を付与するのは、ミスしたブロックのデータを単純に外部ブロックバ

ッファ 110 を介してキャッシュメモリ 2 内に記憶する方法では、複数のミスヒットの発生に対応できないからである。

【0052】ブロックがダーティであるかクリーンであるかは、次に述べるライトオペレーションによる。

【0053】(c) ライトオペレーション (ヒット時) 最後にライトオペレーションについて説明する。本実施例では、いわゆるライトバック法を採用しているが、本発明はライトバック法に限らず、ライトスルーにも適用できる。

【0054】ライト要求はアドレス、ライトデータおよびスレッド識別子と共に、アドレスデコーダ 11、12 および各パイプラインラッチ 20、21、22 を通じて、タグメモリアレイ 31~34 に送られる。そしてタグアクセスの後、比較器 35、36 によってタグ比較が行なわれるところまではリードオペレーションと全く同様である。

【0055】キャッシュメモリのデータメモリアレイ 41~44 には、通常、ブロック毎に 1 ビットのダーティのビットフィールドが設けられている (図示せず)。いづれかの比較器 35、36 の出力が「等しい」とき、すなわちヒットの時は、ライトデータとアドレスとをデータメモリアレイ 41~44 に送り書き込みを行なうと共に該当ブロックのダーティビットをオンにする。

【0056】(d) ライトオペレーション (ミスヒット時)

いづれの比較器 35、36 の出力も「等しくない」とき、すなわちキャッシュメモリ 2 に該当ブロックが存在せずミスヒットのときは、該当ブロックの存在する主メモリ 3 のみにライト要求を出して書き込みを行なう。

【0057】この方法はいわゆるライト・ノー・アロケート法として知られている。主メモリ 3 への書き込みはプロセッサ 1 の動作と並行して行なわれるので、ライトミスのときは、ミスを生じたスレッドを停止する必要はない。ライトミスをしたブロックはキャッシュには存在しないので、後のリードアクセス時には必ずミスヒットとなり、前述のリードオペレーションにより、キャッシュメモリ 2 に読み込まれる。

【0058】

【発明の効果】以上説明したように、本発明によれば、例えばマルチスレッドプロセッサなどに好適なパイプライン構造を有するキャッシュメモリにおいて、例えばキャッシュアクセスがミスヒットの場合でも、キャッシュエントリの更新処理と後続のスレッドからのアクセス処理とを同時並行に行なうことが可能となる。そして、メモリアレイへのアクセスが競合してパイプライン動作が乱れ、キャッシュメモリのスループットが低下するというような問題が起ることはない。

【0059】ここに記載された本発明の実施の形態は単なる一例であり、上記キャッシュメモリ更新回路の実施

の形態は、本発明の要旨を変更しない範囲で多様に変形することができる。

【図面の簡単な説明】

【図1】本発明のパイプラインキャッシュの更新回路を示す図である。

【図2】パイプラインキャッシュとプロセッサおよび主メモリとの間で伝送される信号の構成を示す図である。

(a) メモリシステムの構成、(b) パイプラインキャッシュとプロセッサ間、(c) パイプラインキャッシュと主メモリ間

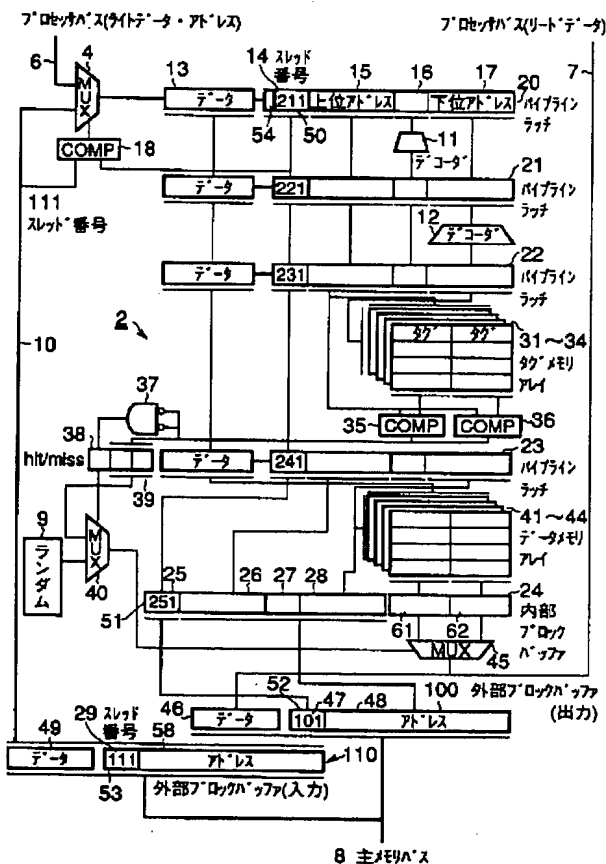
【図3】パイプラインの各ステージにおける処理内容を示す図である。

(a) 読み出し（ヒット）、(b) 読み出し（ミスヒット）、(c) 書き込み、または更新

【符号の説明】

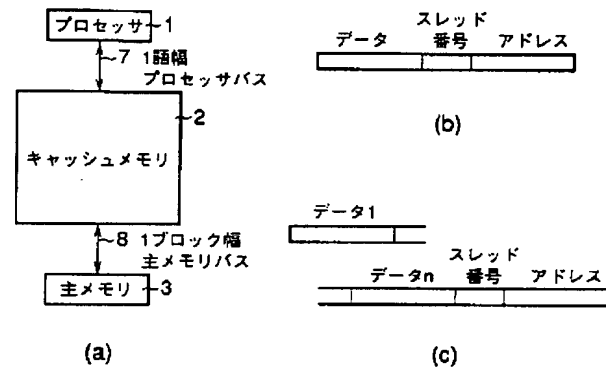
- 1 … プロセッサ
- 2 … キャッシュメモリ
- 3 … 主メモリ
- 4、40、45 … マルチプレクサ
- 6、7、8、10 … バス
- 9 … ランダム発生器
- 11、12 … デコーダ

【図1】



- 13、46、49 … データフィールド
- 14、25、29、47 … スレッド識別子（スレッド番号で表示）
- 15、26 … 上位アドレスフィールド
- 16、17、27、28 … 下位アドレスフィールド
- 18、35、36 … 比較器
- 20、21、22、23 … パイプラインラッチ
- 24 … 内部ブロックバッファ
- 31、32、33、34 … タグメモリアレイ
- 37 … NAND回路
- 38 … ヒット／ミスフィールド
- 39 … 比較結果フィールド
- 41、42、43、44 … データメモリアレイ
- 48、58 … アドレスフィールド
- 50、51、52、53 … スレッド識別子格納フィールド
- 54 … 空きサイクル表示フィールド
- 61、62 … データフィールド
- 100、110 … 外部ブロックバッファ
- 211、221、231、241、251、101、11 … スレッド番号

【図2】



【図3】

